

# COMP 9 / EN47 - Exploring Computer Science: Introduction

## Goals

- Gain an appreciation for the diverse areas encompassing modern computer science
- Understand that computers have limitations, not just due to hardware but even in *theory* (there is no ‘magic’)
- Obtain a working command of the Ruby programming language, and the ability to simplify tedious tasks programmatically
- For example, given a spreadsheet of data that needs somehow transformed and reports produced, write a Ruby program to do the transformations in a way that is *reusable*, rather than doing it by hand or in Excel
- Begin to learn computational problem-solving skills

## Who should take this course?

Anyone! If you wish to make computers less mystifying, spend less time doing tedious tasks, are thinking about studying computer science further, or are just curious, this is the course for you.

**“Computer Science is no more about computers than Astronomy is about telescopes” - Edsger Dijkstra**

## Areas of Computer Science Research at Tufts

- Machine Learning
- Computer Vision
- Graphics
- Programming Languages
- Robotics
- Human-Computer Interaction
- Computation Theory
- Computational Biology

- Computer Architecture
- Algorithms
- Artificial Intelligence
- Operating Systems
- Computational Geometry

## **What is a computer? (group activity, then discussion)**

### **Computational problem solving**

- Identify a problem
- Come up with an elegant (simple, reusable, general) solution
- Implement (automate) that solution

This allows us to turn our problems into a computer's problems, and frees humans to solve interesting, rather than tedious, problems.

“Computer FAIL”

- Performing tedious manual tasks on a computer without taking advantage of automation
  - Reversing first and last names in a text document
  - Changing capitalization (case) of the first word of every sentence.
  - Gathering price information from online book retailers to find the lowest prices (Getchabooks!)

### **What is an algorithm?**

“A specific, step-by-step plan for a procedure that begins with an input value and yields an output value in a finite number of steps”

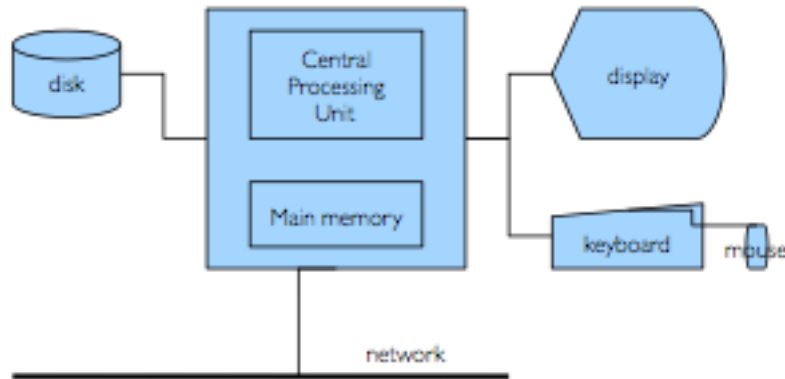


Figure 1: Computer Architecture

## Computers and computers

The term ‘computer,’ or more correctly ‘digital electronic computer’ comes from early 20th century nomenclature: a ‘computer’ was a person who performed computations (financial, codebreaking, etc.) by hand

A modern computer has an architecture something like this:

- Input is how information gets to the computer
  - (the problems provided to the early-20th-century ‘computer’)
- Output is how information gets out of the computer
  - (the report provided by that human computer to someone else)
- Central Processing Unit: this is the part that decodes and executes instructions, carrying out a computer program
  - (the rules the human ‘computer’ learned for performing multiplication or division)
- Memory (Random Access Memory, RAM) is where intermediate results are stored
  - (Scratch paper on which to carry ones, remember quotients, and so on)

- Disk (stable storage) is where long-term results are stored
  - (Filing cabinet for storing the results of computations for later reference)
- The ‘bit’ (**binary digit**) is the fundamental unit of information: 0 or 1
  - (no obvious human analogy without going into fairly deep mathematics and information theory)
  - Used for digital electronic computers because it’s convenient to represent ‘on’ and ‘off’ with electricity
  - Ultimately far deeper; has meaning in information theory even without electronics
- The ‘byte’ is just a convenient level of organization, 8 bits
  - Kilobyte, Megabyte, Gigabyte: thousands, millions, or billions of bytes

## Programming

Let’s make a sandwich! Explain how?

Programming is translating *English* (or another natural language) into an *algorithm* and finally into a *program*

### The Ruby programming language

- Ruby is a modern programming language, designed by Yukihiro Matsumoto (“Matz”) and first released in 1995.
- Matz designed Ruby with the goal of increasing programmers’ happiness and productivity, and putting the needs of programmers above those of computers.
- Ruby is used heavily in industry (a great deal of web application development takes place in Ruby) but we are using it here because it will allow you to go further in this exploration of computer science than if we were using a language like C++ or Java.
- Ruby makes programming easier and more fun, so we can focus on the really exciting parts of computer science.

## **Administrivia**

### **Create an account**

[www.cs.tufts.edu/~accounts/](http://www.cs.tufts.edu/~accounts/)

### **Assignment 0 (due Tuesday, January 25th in class)**

Write a few paragraphs (and not more than a page) telling me about yourself. What are your interests, both inside and outside of academics? What is your major, if you've decided, and otherwise, what are you thinking about majoring in? Do you have any career goals? If so, what are they? Do you have any experience programming or playing with computers, or are you absolutely terrified of them? What do you hope to get out of this course?

This assignment will not be given a letter grade, but completion of it counts for 5% of your grade for the course. Your motivation for this is that I will use your answers to design the second half of the course; the topics we discuss will attempt to reflect your interests.

Please submit a written (preferably typed) page, hard copy (on paper) in class on Tuesday. Future assignments will be submitted electronically.